



Fault-Tolerant Softcore Processors

Part I: Fault-Tolerant Instruction Memory



Nathaniel Rollins

Brigham Young University



THE GEORGE
WASHINGTON
UNIVERSITY
WASHINGTON DC



VIRGINIA POLYTECHNIC INSTITUTE
AND STATE UNIVERSITY



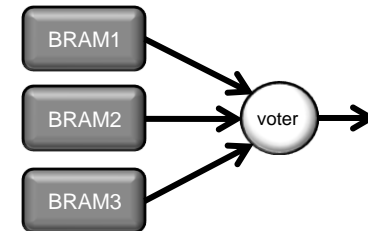
Overview

- Strong interest in FT softcore processors in space
 - LEON processor used by European space program
 - Microblaze, PicoBlaze, 8051, ERC32, etc.
- Rad-hard processors are **expensive, big, and slow**
- Softcore processors are **flexible, fast, and cheap**

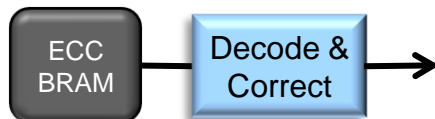
- **Overall Goal**: identify low cost SEU mitigation techniques for softcore processors
 - **Goal of Part I study**: Identify low cost SEU mitigation techniques for softcore processor instruction memories

Approach

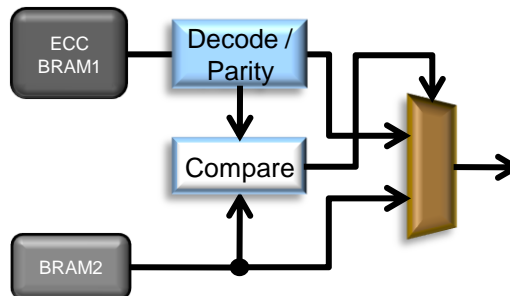
- **TMR** is the most common mitigation technique
 - **Expensive and slow**
- Other hardware techniques
 - **Detection isn't good enough – must correct**
 - DWC alone isn't good enough
 - EDC alone isn't good enough



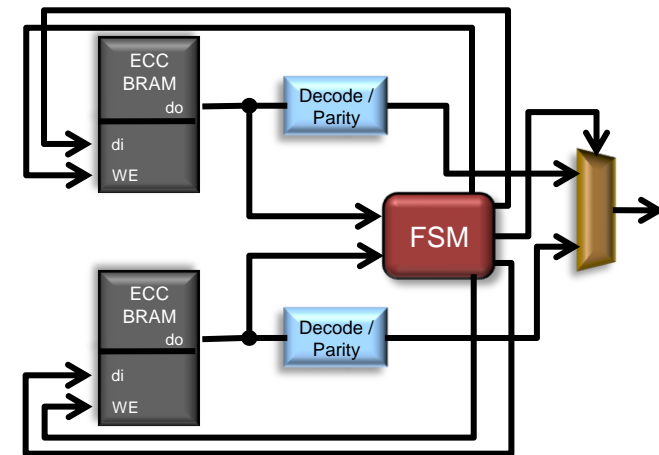
ECC



EDC with DWC



Scrubbing

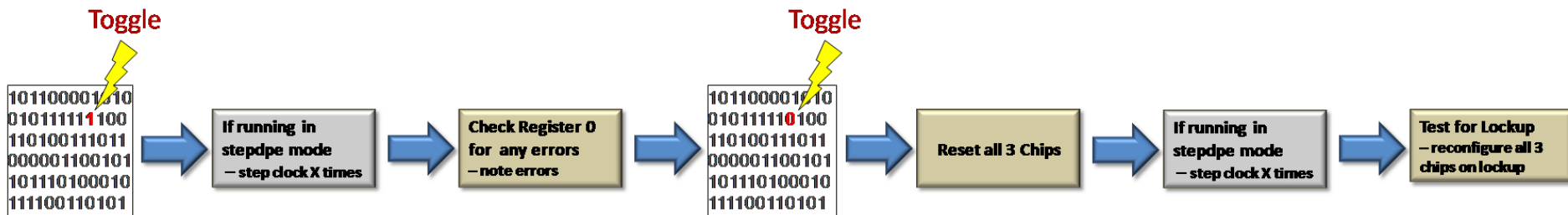
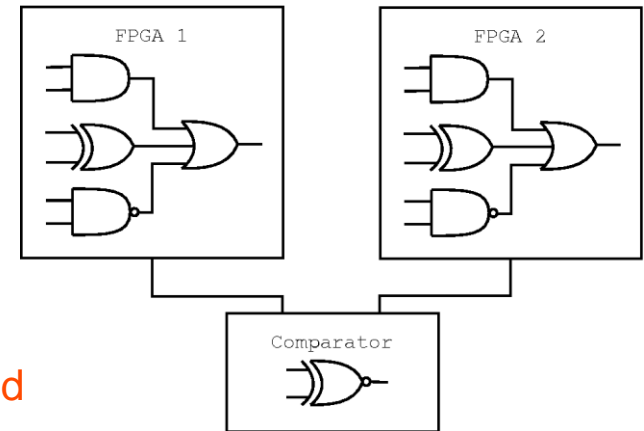


- **Study Approach**

- **Compare different software processor instruction memory fault-tolerant techniques in terms of:**
 - **Area, speed, power, reliability**
- **Remaining processor protection: plain TMR**

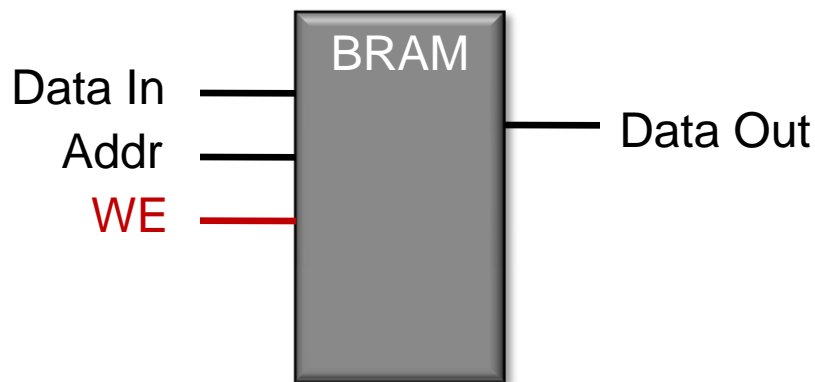
Fault Model

- BYU/LANL SLAAC1V fault injection tool used to insert single bit upsets into Virtex FPGAs
 - BRAM bits in Virtex bitstream are treated differently
 - **Task:** upgrade fault injection tool to support:
 - Upsets in BRAM
 - Readback of BRAM bits
- Next studies use SEAKR XRTC board with Virtex4 FPGA
 - SEAKR board borrowed from LANL
 - Fault injection tool also upgraded to upset BRAMs and detect critical failures



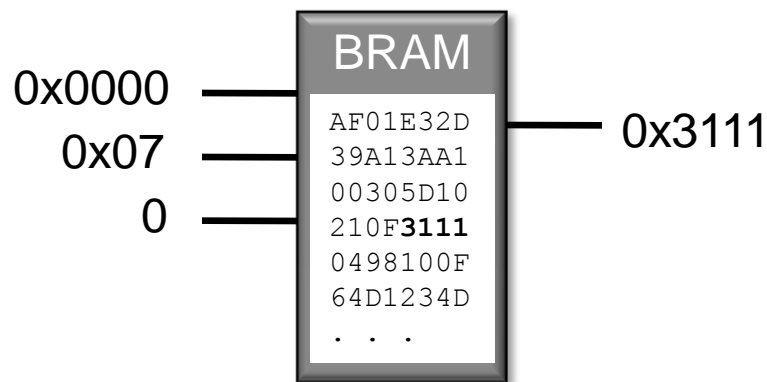
Critical Failures

- **Critical Failures**: upsets that cannot be fixed with a reset (lead to a SEFI)
 - Different memory structures are susceptible to critical failures:
 - BRAMs
 - LUTRAMs
 - SRLs
 - Registers that are not tied to a global reset
 - Example: WE port on a BRAM



Critical Failures

- **Critical Failures**: upsets that cannot be fixed with a reset (lead to a SEFI)
 - Example: WE port on a BRAM

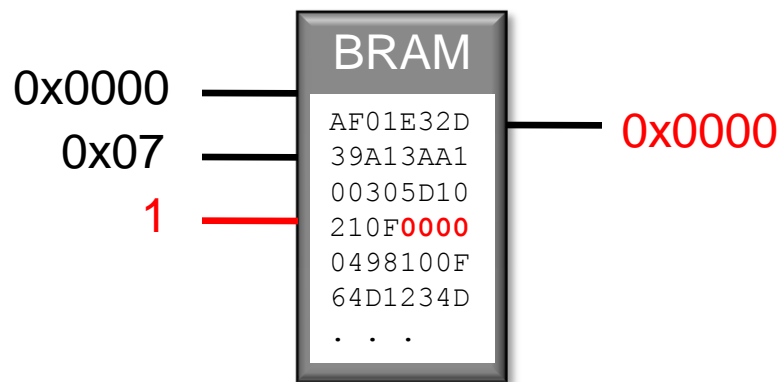


Instruction memory should never be written to → BRAM is treated as a ROM

- input data lines tied low
- WE tied low

Critical Failures

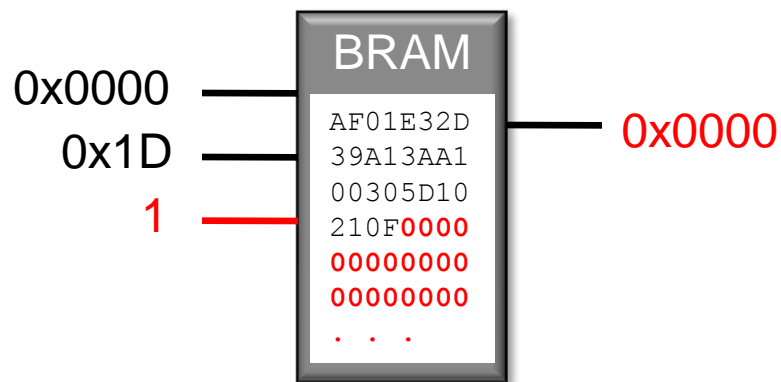
- **Critical Failures**: upsets that cannot be fixed with a reset (lead to a SEFI)
 - Example: WE port on a BRAM



Upsetting the **WE** port overwrites the BRAM contents

Critical Failures

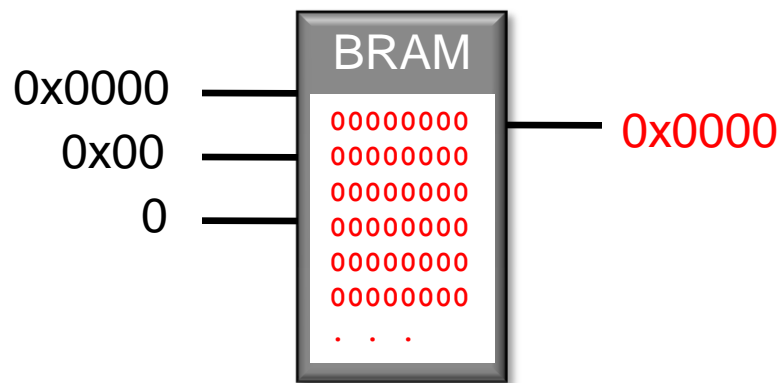
- **Critical Failures**: upsets that cannot be fixed with a reset (lead to a SEFI)
 - Example: WE port on a BRAM



Especially bad for processors since BRAM address continually increments

Critical Failures

- **Critical Failures**: upsets that cannot be fixed with a reset (lead to a SEFI)
 - Example: WE port on a BRAM

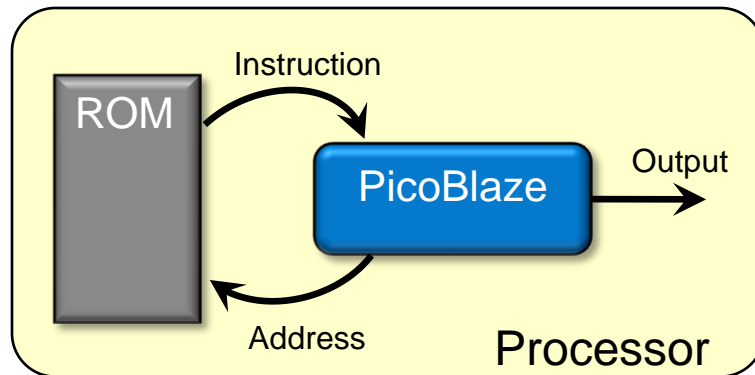


Resetting the device will restart the processor, but will not restore the BRAM contents (program is lost)!

- Mitigation techniques need to eliminate critical failures

Fault-Tolerant Techniques

- Original processor design: Xilinx PicoBlaze



Fault-tolerance determined by examining the **PC** and current **instruction** as faults are injected

- Instruction memory fault-tolerant techniques:

- **TMR:**

- Single voter
- Triple voter
- Feedback
- BLTMR
- Scrubber

- **ECC:**

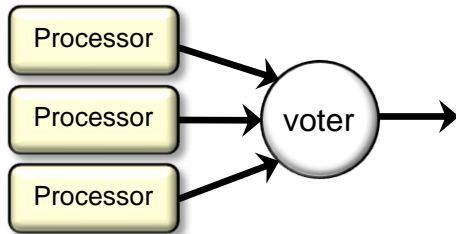
- SEC/DED
- SEC/DED with DWC
- SEC/DED with DWC and scrubbing

- **EDC & DWC:**

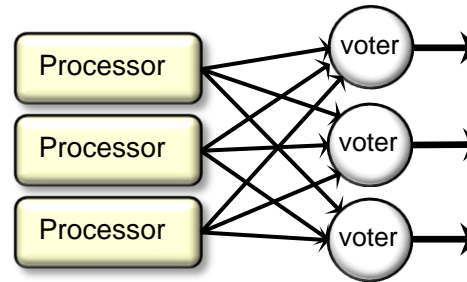
- CD with DWC
- CD with DWC and scrubbing

Fault-Tolerant Techniques: TMR

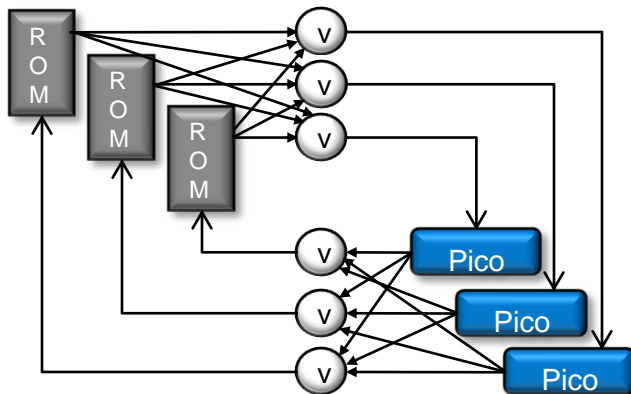
Top-Level TMR – 1 voter



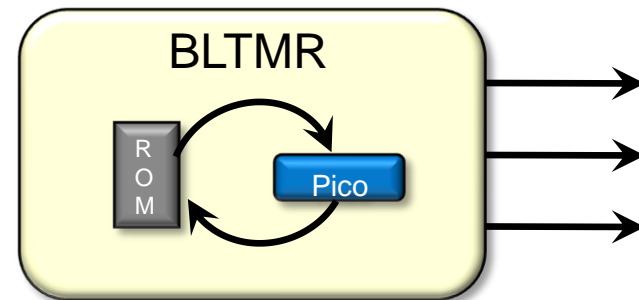
Top-Level TMR – 3 voters



Feedback TMR

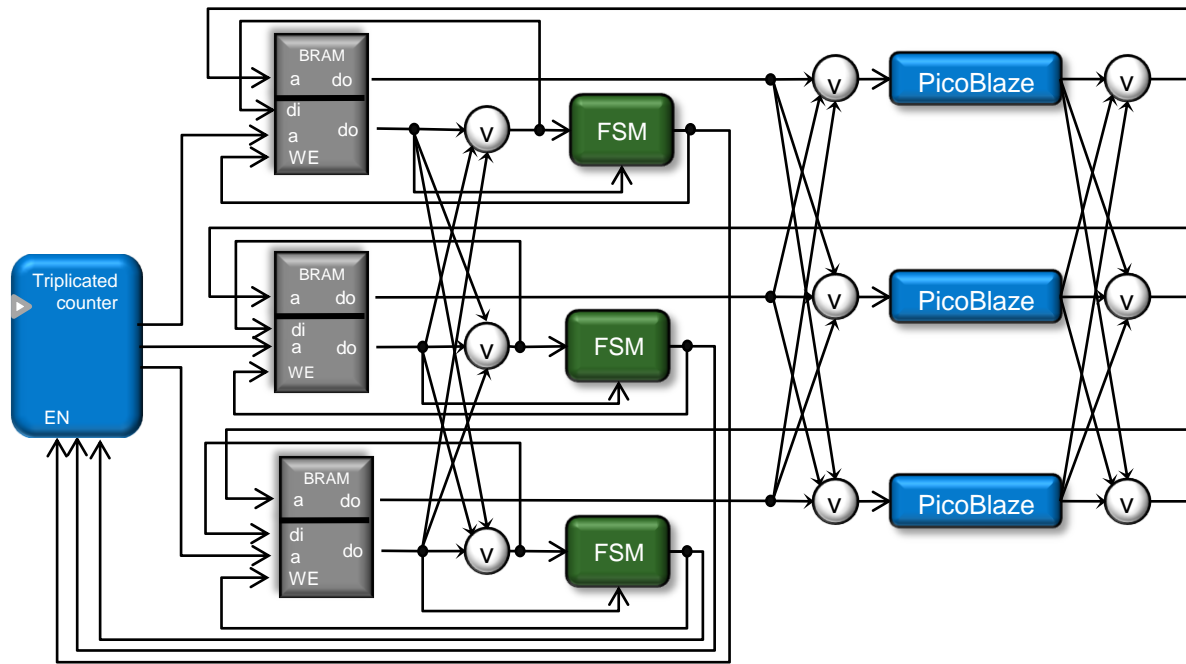


BYU/LANL TMR Tool



FT Techniques: TMR with Scrubbing

- BYU/Sandia BRAM scrubber with TMR
 - Each BRAM scrubbing WE must be independent of other BRAM WEs
 - Scrubbing address counters MUST be kept in sync
 - Scrubbing counter must be 2x slower than BRAM clock
 - Must prevent read/write address conflicts

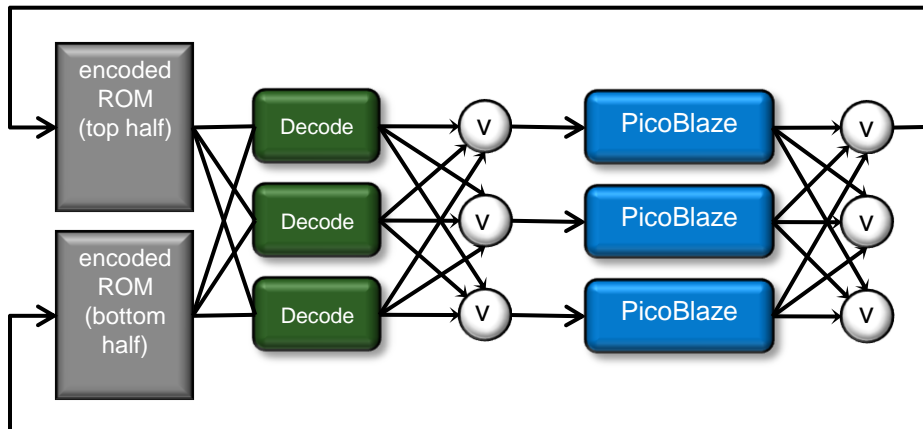


Without scrubbing overlapping errors will cause TMR to fail

Eliminating critical failures is difficult when BRAM WEs are upset

FT Techniques: SEC/DED

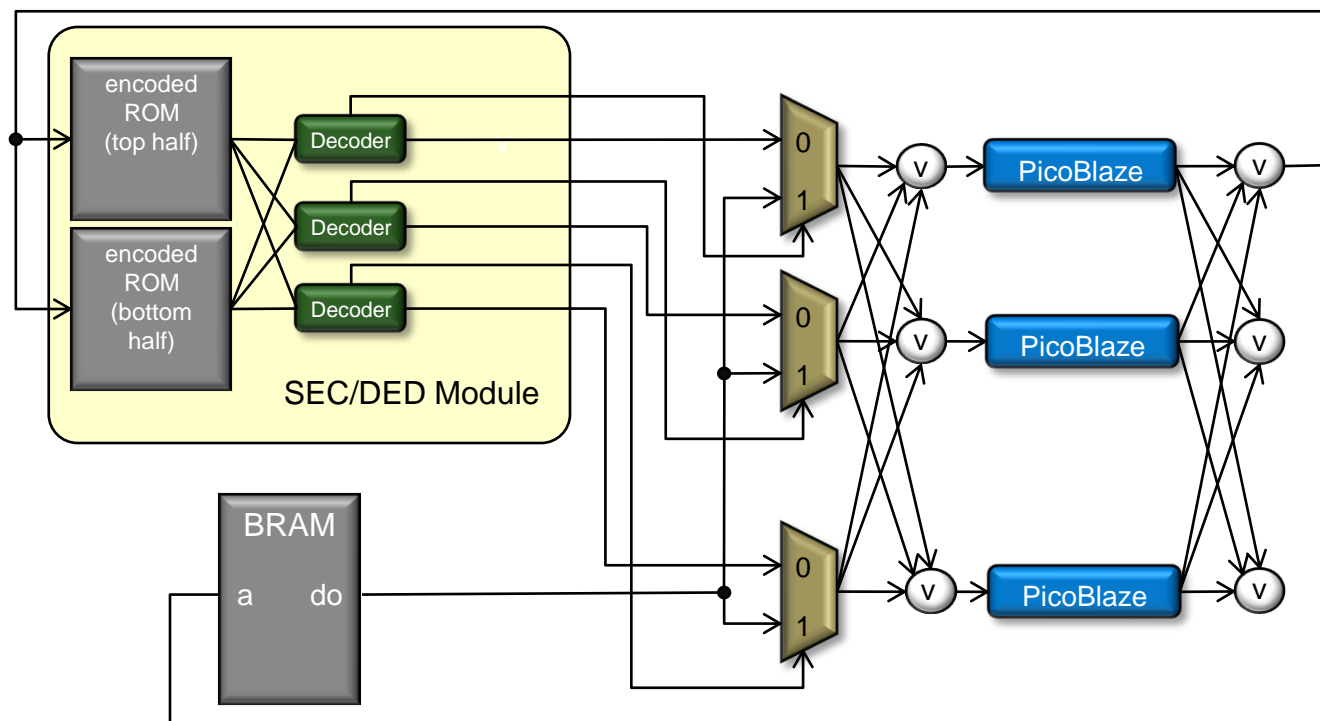
- SEC/DED on 16-bit word:
 - Use (22, 6) code on 16-bit word
 - Use 2 BRAMS:
 - 1 for top half encoded word (11 bits)
 - 1 for bottom half encoded word (11 bits)
- Complete fault tolerance difficult when crossing from triplicated to non-triplicated
 - Logic and routing coming into and out of BRAMs are single point of failure



- SEC/DED:
 - Detects and corrects any single-bit upset
 - Detects any double-bit upset
 - Triple+ upsets may or may not be detected

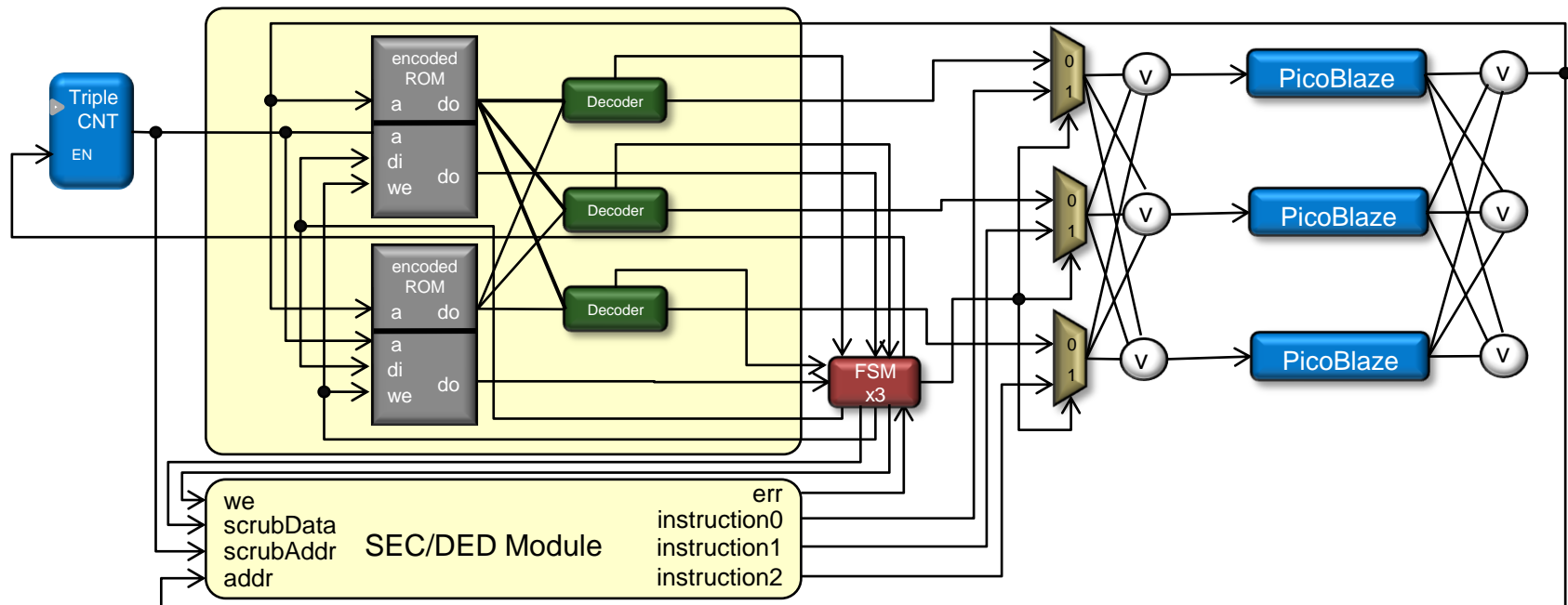
FT Techniques: SEC/DED with DWC

- Improve SEC/DED reliability with DWC
- Still susceptible to critical failures when BRAM WE is upset



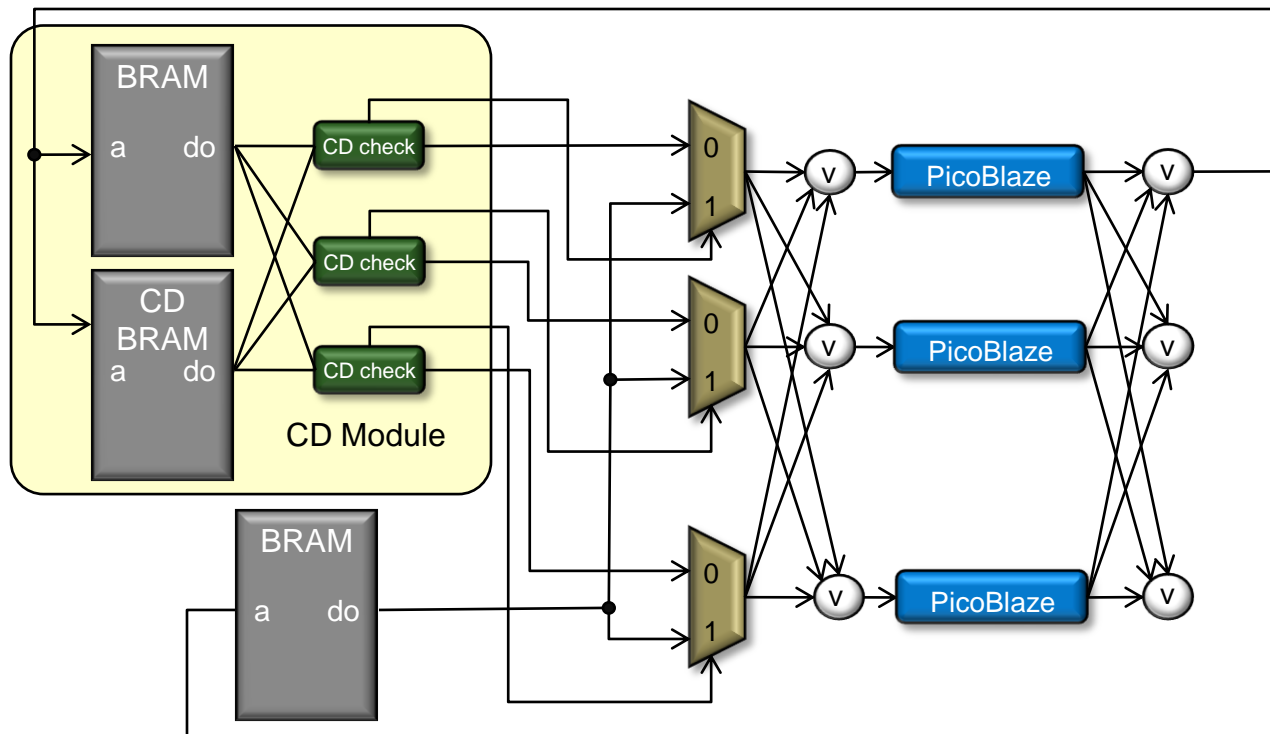
FT Techniques: SEC/DED DWC Scrub

- Scrubbing uses dual ported BRAMs
 - Scrub address counter runs $\frac{1}{2}$ speed of BRAM clock
- Scrubbing cannot fix all errors (only single-bit/double-bit guaranteed)
 - **Scrub trigger**: single error correction(SEC) or double error detection (DED) on current instruction – more than 2 errors may or may not be caught
 - When triggered, a scrub copies entire BRAM contents of good BRAM into bad BRAM



FT Techniques: CD with DWC

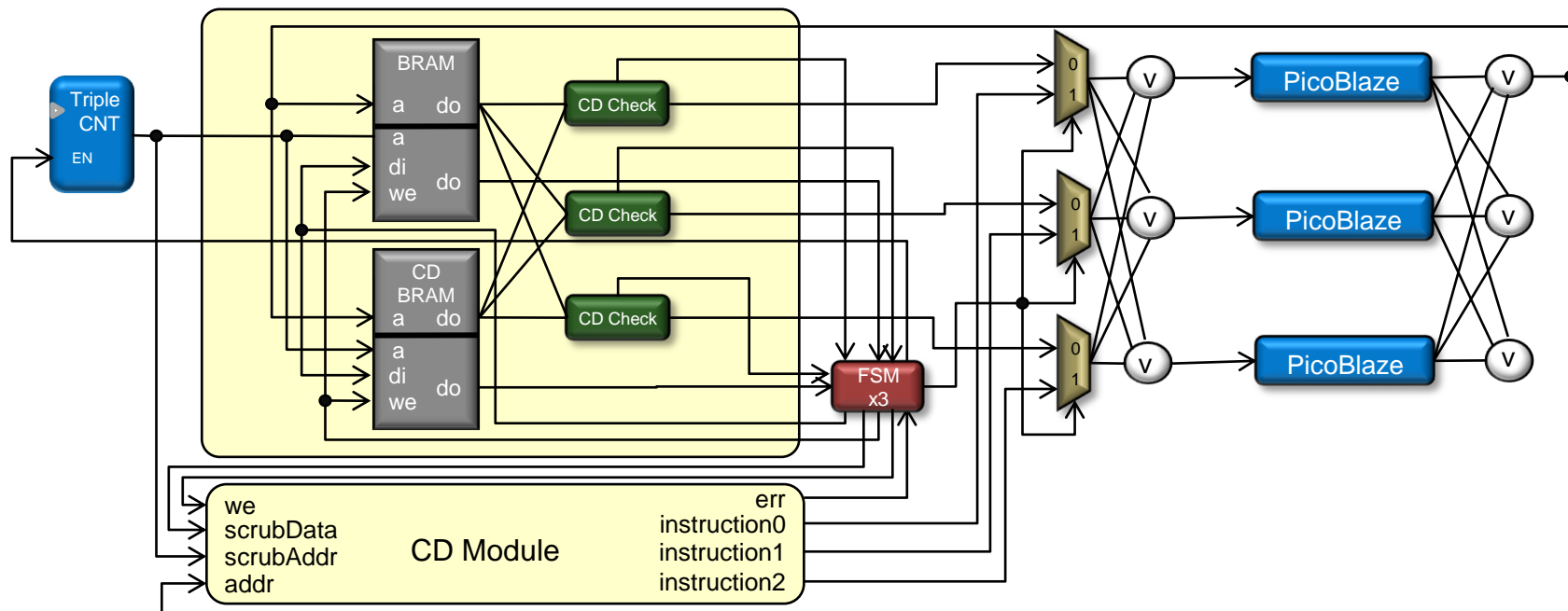
- Complement Duplicate (CD) duplicates and inverts (complements) the original BRAM contents
 - Detects errors by comparing the original with the complemented CD
- CD only **detects** upsets so DWC is used to **correct** upsets



- **CD detects:**
 - Any single-bit upset
 - 66% double-bit upsets
 - Any multiple adjacent unidirectional upset

FT Techniques: CD DWC Scrub

- Scrubbing uses dual ported BRAMs
 - Scrub address counter runs $\frac{1}{2}$ speed of BRAM clock
- Scrubbing will fix critical failures
 - **Scrubbing trigger**: inverse of current instruction doesn't match CD contents
 - When triggered, a scrub copies entire BRAM contents of good BRAM into bad BRAM
 - There are other scrubbing design strategies with CD – but this one removes all critical failures



FT Techniques: Results

Design	Slices		BRAM Bits		Clock Rate (MHz)		Power (mW)		Sensitive Bits		Critical Failures
Original	70		560		65.5		49		2881		3
1 voter	227	3.2x	1680	3x	67.5	1.03x	66	1.35x	847	3.4x	3
3 voters	252	3.6x	1680	3x	71.4	1.09x	75	1.53x	36	80.0x	3
Feedback	250	3.6x	1680	3x	66.1	1.01x	73	1.49x	68	42.4x	3
BLTMR	297	4.2x	1680	3x	63.9	1.03x	76	1.55x	52	55.4x	3
TMR Scrub	348	5.0x	1680	3x	58.4	1.12x	82	1.67x	28	102.9x	0
SEC/DED	340	4.9x	770	1.4x	43.4	1.51x	82	1.67x	711	4.1x	16
SEC/DED DWC	373	5.3x	1540	2.8x	42.7	1.53x	89	1.82x	473	6.1x	3
SEC/DED DWC Scrub	545	7.8x	1540	2.8x	32.4	2.02x	105	2.14x	326	8.8x	0
CD DWC	235	3.4x	2240	4x	47.9	1.37x	72	1.47x	1034	2.8x	2
CD DWC Scrub	395	5.6x	2240	4x	29.7	2.21x	90	1.84x	231	12.5x	0

Clock and reset lines are NOT triplicated

Conclusions

■ Reliability

- For instruction memories, **TMR with scrubbing** provides the best protection
 - Fewest sensitivities
 - Eliminates critical failures
- Scrubbing is required to eliminate critical failures

■ Costs

- **TMR** is more effective than **SEC/DED** and **CD with DWC**
 - Better protection
 - Lower area, speed, and power costs
- **SEC/DED** and **CD with DWC** scrubbers are very expensive

FT Softcore Processors: Moving Forward

- Next General Studies:
 - Memory Study: BRAMs & LUTRAMs
 - Software fault-tolerant techniques study
- Create different fault models for SEAKR board
 - Multi-bit upset model
 - Temporal fault-tolerant techniques model
- Combinations of different fault-tolerant techniques

